IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **Nenov et al.**	Examiner: Behzad Peikari
Application No: 10/849,354	Art Unit: 2189
Filed: May 18, 2004	Confirmation No: 6296
For: HYBRID CACHE HAVING STATIC AND DYNAMIC PORTIONS	

Mail Stop Amendment
Commissioner For Patents
P.O. Box 1450
Alexandria, VA 22313-1450

In response to the Office Action mailed April 18, 2007, Applicants respectfully request reconsideration of the above-referenced U.S. Patent application in light of the following.

FILED BY EFS WEB June 18, 2007

SPECIFICATION AMENDMENTS

Please replace paragraph [0020] with the following amended paragraph:

[0020] Throughout this specification, several terms of art are used. These terms are to take on their ordinary meaning in the art from which they come, unless specifically defined herein or the context of their use would clearly suggest otherwise. A "cache" is defined herein as a buffer used to speed up data retrieval. A cache may be established ~~**with in**~~ **within** volatile memory (e.g., random access memory) or within non-volatile memory (e.g., hard disk).

Please replace paragraph [0022] with the following amended paragraph:

[0022] Hybrid-cache 110 caches (i.e., buffers) data received from one or more data sources 130. In turn, hybrid-cache 110 may provide the cached data to data consumers 135 more quickly than if the data was directly obtained from data sources 130. As such, the more data buffered within hybrid-cache 110, the more likely **[[a]]** requests for data issued by data consumers 135 may be quickly serviced from cached copies of the data within hybrid-cache 110, as opposed from data sources 130. Finding a match between a request for data and cached data is referred to as a "cache hit." The more likely a cache hit occurrence, the overall performance of processing system 100 is increased and data is provided to data consumers 135 in a more timely manner.

Please replace paragraph [0027] with the following amended paragraph:

[0027] Embodiments of the present invention are capable of implementing a variety of different eviction policies for both static cache 120 and dynamic cache 125. Both static cache 120 and dynamic cache 125 may implement the same eviction policy or different evictions policies, according **to** the task at hand. Similarly, embodiments including multiple dynamic caches may include dynamic caches implementing the same eviction policy or different eviction policies.

Please replace paragraph [0029] with the following amended paragraph:

[0029] Referring to FIG. 2, data cached within static cache 120 is called "stable data" and data cached within dynamic cache 125 is called "soft data." The data cached within static cache 120 is referred to as stable data because it is less likely to be completely evicted from hybrid-cache 110, than data cached within dynamic cache 125. The data cached within dynamic cache 125 may be said to be "softly reachable" because it is less likely to remain cached than the data cached in static cache 120. The stable data and soft data may represent ~~**nay**~~ **any** type of cacheable data, including objects of an object orientated language (e.g., Java), database files, archive files, application files, data files, and the like.

Please replace paragraph [0041] with the following amended paragraph:

[0041] If the demand for memory 105 exceeds the supply of free or available memory 105 (decision block 510), then dynamic cache 125 may need to contract to free up memory 105 for other uses, such as applications 115. Without contracting dynamic cache 125, applications 115 will not be able to expand and **[[an]]** **a** stack overrun or insufficient memory error may occur, else in some embodiments, applications 115 may need to be swapped to a hard disk of processing system 100, considerably slowing executing of applications 115. In this scenario, process 500 continues to a process block 520.

LISTING OF CLAIMS

1. (**Currently Amended**) A caching method, comprising:

caching data received from a data source within a static cache as stable data, the static cache having a fixed size;

evicting a portion of the stable data within the static cache to a dynamic cache when the static cache reaches a threshold fill level; and

enrolling the evicted portion of the stable data into the dynamic cache as soft data, the dynamic cache having a dynamically changing size **according to availability of memory, where soft data is evicted from the dynamic cache prior to evicting stable data from the static cache if availability of memory is scarce**.

2. (**Canceled**)

3. (**Currently Amended**) The caching method of ~~**claim 2**~~ **claim 1**, wherein evicting the portion of the stable data further comprises evicting the portion of the stable data to the dynamic cache according to a Least Recently Used eviction policy.

4. (Previously Presented) The caching method of claim 1, further comprising:

evicting selectively at least some of the soft data from the dynamic cache when the availability of the memory is scarce; and

contracting the dynamic cache to release some of the memory consumed by the dynamic cache.

5. (Original) The caching method of claim 4, wherein evicting selectively the at least some of the soft data further comprises evicting the at least some of the soft data according to a Least Recently Used eviction policy.

6. (Previously Presented) The caching method of claim 4, wherein enrolling the evicted portion of the stable data into the dynamic cache as soft data comprises caching the soft data as hash values of a hash table, the hash values being indexed to keys for accessing the hash values.

7. (Original) The caching method of claim 6, wherein evicting selectively at least some of the soft data from the dynamic cache comprises:

copying at least some of the keys into a garbage queue, the at least some of the keys corresponding to the at least some of the soft data; and

removing at least some of the hash values from the hash table based on the at least some of the keys in the garbage queue.

8. (Original) The caching method of claim 7, wherein a Java Garbage Collector selectively copies the at least some of the keys into the garbage queue.

9. (**Currently Amended**) The caching method of ~~**claim 1**~~ **claim 2**, wherein the data comprises first data, the method further comprising:

intercepting a request for second data from the data source;

determining whether the second data is cached within either of the static cache and dynamic cache; and

providing the second data from either of the static cache and the dynamic cache instead of the data source, if the determining determines that the second data is cached.

10. (Previously Presented) The caching method of claim 9, further comprising moving the second data to a most recently used position within the static cache, upon determining that the second data is cached.

11. (**Currently Amended**) The caching method of ~~**claim 1**~~ **claim 2**, wherein the static cache and the dynamic cache comprise a hybrid-cache within a single memory device.

12. (**Currently Amended**) The caching method of ~~**claim 1**~~ **claim 2**, wherein the stable data and the soft data comprise objects of an object orientated language.

13. (Currently Amended) [[A]] **An article of manufacture comprising a** machine-accessible medium ~~**that provides**~~ **having** instructions **stored thereon** that, if executed by a machine, will cause the machine to perform operations comprising:

caching first data received from a data source into a hybrid-cache, the hybrid-cache including a static cache having a fixed size and a dynamic cache having a dynamically changing size;

enrolling the first data received from a data source into the static cache as stable data;

evicting selective portions of the stable data within the static cache to the dynamic cache when the static cache is full; and

enrolling the selective portions of the stable data evicted from the static cache into the dynamic cache as soft data**, the dynamic cache having a dynamically changing size according to availability of memory, where soft data is evicted from the dynamic cache prior to evicting stable data from the static cache if availability of memory is scarce**.

14. (Canceled)

15. (Currently Amended) The ~~**machine-accessible medium**~~ **article of manufacture** of claim 13, further providing instructions that, if executed by the machine, will cause the machine to perform further operations, comprising:

expanding the dynamic cache to accommodate the selective portions of the stable data evicted to the dynamic cache, if adequate memory is available; and

evicting at least some of the soft data from the dynamic cache to accommodate the selective portions of the stable data evicted to the dynamic cache, if adequate memory is not available.

16. (Currently Amended) The ~~**machine-accessible medium**~~ **article of manufacture** of claim 15, further providing instructions that, if executed by the machine, will cause the machine to perform further operations, comprising:

contracting the dynamic cache to release some of the memory consumed by the dynamic cache, if the memory is scarce.

17. **(Currently Amended)** The ~~machine-accessible medium~~ **article of manufacture** of claim 15, wherein enrolling the selective portions of the stable data evicted from the static cache into the dynamic cache as the soft data comprises caching the soft data within the dynamic cache according to a canonical mapping scheme.

18. **(Currently Amended)** The ~~machine-accessible medium~~ **article of manufacture** of claim 17, wherein caching the soft data within the dynamic cache according to the canonical mapping scheme comprises caching the soft data as a hash value of a hash table, the hash values being indexed to keys for accessing the hash values.

19. **(Currently Amended)** The ~~machine-accessible medium~~ **article of manufacture** of claim 18, wherein evicting the at least some of the soft data from the dynamic cache comprises:

copying at least some of the keys into a garbage queue, the at least some of the keys corresponding to the at least some of the soft data; and

removing at least some of the hash values from the hash table based on the at least some of the keys in the garbage queue.

20. **(Currently Amended)** The ~~machine-accessible medium~~ **article of manufacture** of claim 13, wherein evicting selective portions of the stable data within the static cache comprises evicting the selective portions of the stable data according to a Least Recently Used eviction policy.

21. **(Currently Amended)** The ~~machine-accessible medium~~ **article of manufacture** of claim 13, wherein the stable data and the soft data comprise objects of an object orientated language.

22. **(Currently Amended)** A system, comprising:

a processor to process requests for data from a data source; and

a memory device communicatively coupled to the processor, the memory device to hold a hybrid-cache, the hybrid-cache comprising:

a static cache for caching the data as stable data, the static cache having a fixed size; and

a dynamic cache having a dynamically changing size according to availability of memory within the memory device, wherein portions of the stable data within the static cache are to be evicted to the dynamic cache as soft data when the static cache is full**, wherein the dynamic cache is to expand to accommodate the portions of the stable data evicted to the dynamic cache when the static cache is full, if adequate memory is available within the memory device, and wherein the dynamic cache is further to evict at least some of the soft data from the dynamic cache to accommodate the portions of the stable data evicted to the dynamic cache, if adequate memory is not available within the memory device, where soft data is evicted from the dynamic cache prior to evicting stable data from the static cache if availability of memory is scarce**.

23-24. (Canceled)

25. (**Currently Amended**) The system of ~~**claim 24**~~ **claim 22**, wherein the dynamic cache is further to contract to release memory consumed by the dynamic cache, if other entities within the memory device expand.

26. (**Currently Amended**) The system of ~~**claim 24**~~ **claim 22**, wherein the memory device comprises Random Access Memory ("RAM") and wherein the data source comprises a data storage device communicatively coupled to the processor, the hybrid-cache to reduce swapping to the data storage device.

27. (Previously Presented) The system of claim 22, wherein the system comprises a caching server, wherein the requests for the data from the data source comprise requests from clients of the caching server, and wherein the data source comprises an Internet.

28. (Previously Presented) The system of claim 22, wherein the system comprises an Application Server, wherein the requests for the data from the data source comprise requests from clients of the Application Server, and wherein the data source comprises at least one database.

29. (Original) The system of claim 22, wherein the Application Server comprises one of a Java based Application Server and a .NET based Application Server.

30. (**Currently Amended**) A system, comprising:

static means for caching stable data received from a data source within a fixed amount of memory;

first means for selectively evicting portions of the stable data from the static means when the static means is full;

dynamic means for caching soft data within a dynamically changing amount of memory;

second means for evicting the soft data from the dynamic means when the available amount of memory is scarce; and

means for enrolling the portions of the stable data evicted by the means for evicting into the dynamic means as the soft data**, for caching the soft data within the dynamically changing amount of the memory based on an available amount of the memory, and for contracting the dynamically changing amount of memory when the available amount of memory is scarce, the dynamic means having a dynamically changing size according to availability of memory, where soft data is evicted from the dynamic means prior to evicting stable data from the static means if availability of memory is scarce**.

31-33. (Canceled)

34. (Previously Presented) The caching method of claim 1, wherein the threshold fill level comprises a full static cache.

REMARKS

With this Response, claims 1, 3, 9, 11-13, 15-22, 25-26, and 30 are amended. Applicants respectfully request that claims 2, 14, 23-24, and 31-33 be canceled without prejudice. Therefore, claims 1, 3-13, 15-22, 25-30, and 34 are pending.

ALLOWABLE SUBJECT MATTER

Applicants acknowledge that claims 6-8 were found to have allowable subject matter. More particularly, these claims were objected to as being dependent upon rejected base claims, but would be allowable if rewritten in independent form. Applicants respectfully submit that the rejection of independent claim 1, from which these claims depend, is overcome herein, rendering these claims allowable as currently written.

CLAIM REJECTIONS - 35 U.S.C. § 101

Claims 13-21 were rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter. More particularly, these claims were rejected as having a claim scope that may include virtual objects and/or data structures. These claims are amended herein to recite an "article of manufacture comprising a machine-accessible medium having instructions stored thereon," which Applicants submit falls under one of the four statutory categories of claimable subject matter. Therefore, Applicants respectfully request that the rejection of these claims be withdrawn.

CLAIM REJECTIONS - 35 U.S.C. § 102

Claims 1-2, 4, 9, 13-16, 22-25, and 30-34 were rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,860,081 of Herring et al. (hereinafter "Herring"). Applicants respectfully submit that these claims are not anticipated by the cited reference for at least the following reason.

The Office Action at page 4 asserts that "all that is needed to teach the independent claim features ... is a prior art system wherein an L1 cache is of a fixed size (static) and the L2 cache has an adjustable (dynamic) size." This assertion of the Office Action is based on the assumption at page 3 that "cache hierarchies are designed to evict data from a higher level cache to a lower level cache." Applicants must respectfully disagree with this reasoning and the application of

Herring to the claimed invention. There is no support in the reference for the assumption at page 3 of the Office Action. Per MPEP § 2131, "A claim is anticipated only if each and every element as set forth in the claim is found, **either expressly or inherently described**, in a single prior art reference." Case law citations omitted. Applicants point out that the Herring reference is **silent** as to eviction policies, either the eviction of data from L1 or the eviction of data from L2. Thus, the reference fails to expressly describe at least one element as set forth in the claim. Regarding inherently describing the features of the claim, Applicants observe that one of skill in the art discussed in Herring would be familiar with the fact that L1/L2 cache systems are commonly managed so that the L2 cache includes **all** the data that is in the L1 cache. Such a system configuration is not consistent with an assumption that data evicted from a higher level is evicted to a lower level cache, seeing that the lower level cache already had the data to begin with. Thus, Applicants submit that it is improper to assume that the silence of the reference discloses the features of Applicants' claims. There is no teaching either express or inherent in the reference that would suggest data is evicted from a higher-level cache to a lower-level cache, in contrast to what is set forth in Applicants' claims.

In fact, Applicants' claim 1 recites the following:

> caching data received from a data source within a static cache as stable data, the static cache having a fixed size;
>
> evicting a portion of the stable data within the static cache to a dynamic cache when the static cache reaches a threshold fill level; and
>
> **enrolling the evicted portion of the stable data into the dynamic cache as soft data**, the dynamic cache having a dynamically changing size according to availability of memory, where soft data is evicted from the dynamic cache prior to evicting stable data from the static cache if availability of memory is scarce.

Claims 13, 22, and 30 are also independent claims, and recite limitations directed to enrolling stable data evicted from a static cache as soft data into a dynamic cache.

Besides the deficiencies of failing to point to any teaching or suggestion of evicting data from a static cache to a dynamic cache, as recited in the claims, Applicants submit that the reference fails to disclose or suggest that the evicted data is enrolled as soft data. The Office Action asserts at page 3 that "'stable data' and 'soft data' have no intrinsic distinction from each other besides [the] fact that the first is located in the static memory and the second is located in the dynamic memory." Applicants disagree. The Office Action correctly points to paragraph [0029] of Applicants' Specification, which explains that data in the dynamic cache is soft data

that is "softly reachable." Applicants submit that one of skill in the art would recognize soft data that is "softly reachable" to refer to data that has a "soft reference," which prevents immediate eviction of the data from the cache, but will still allow eviction of the data if memory space is scarce. Thus, as described, it will remain in the cache, but is less likely to remain cached than stable data, which generally has a stronger reference.

Applicants thus submit that the independent claims include at least one feature that is not disclosed or suggested by the cited reference. The reference therefore fails to support an anticipation rejection of the independent claims under MPEP § 2131. The remaining claims depend from the independent claims, and so necessarily include the limitations of the independent claims from which they depend. Thus, the dependent claims are patentable over the cited reference for at least the same reasons as the independent claims.

CLAIM REJECTIONS - 35 U.S.C. § 103

Claims 3, 5, 10-12 and 26

These claims were rejected under 35 U.S.C. § 103(a) as being unpatentable over Herring in view of U.S. Patent No. 6,321,235 of Bird (hereinafter "Bird"). Applicants submit that these claims are not rendered obvious by the cited references for at least the following reasons. These claims depend from independent claims discussed above. As shown above, Herring fails to support a rejection of the independent claims, at least for failing to disclose or suggest at least one feature of the invention as recited in the independent claims. Bird is not cited as curing, nor does it cure the deficiencies of Herring set forth above. Whether or not Bird discusses an LRU implementation, the combination of the references fails to disclose or suggest at least the features discussed above. Therefore, whether alone or in combination, the references fail to support a rejection of the independent claims, and thus fail to support a rejection of these dependent claims.

Claim 17

This claim was rejected under 35 U.S.C. § 103(a) as being unpatentable over Herring in view of Bird, and further in view of U.S. Patent Application Publication No. 2003/0105936 of Stakutis et al. (hereinafter "Stakutis"). Stakutis is cited as disclosing a canonical mapping. Whether or not Stakutis discloses a canonical mapping, the reference fails to cure the

deficiencies of Herring and Bird as set forth above. Claim 13 is not rendered unpatentable by either Herring or Bird, or a combination. The combination of those references with the teachings of Stakutis still fails to disclose or suggest at least one feature of the invention as recited in independent claim 13. Therefore, claim 17, which depends from claim 13, is likewise not rendered unpatentable by the references either alone or in combination.

Claim 27-29

These claims were rejected under 35 U.S.C. § 103(a) as being unpatentable over Herring in view of Bird, further in view of U.S. Patent Application Publication No. 2003/0172145 of Nguyen et al. (hereinafter "Nguyen"). Nguyen is cited as disclosing a caching server. Whether or not Nguyen discloses a caching server, the reference fails to cure the deficiencies of Herring and Bird as set forth above. Independent claim 22, from which these claims depend, is not rendered unpatentable by either Herring or Bird, or a combination. The combination of those references with the teachings of Stakutis still fails to disclose or suggest at least one feature of the invention as recited in independent claim 22. Therefore, these claims are likewise not rendered unpatentable by the references either alone or in combination.

CONCLUSION

For at least the foregoing reasons, Applicants submit that the rejections have been overcome. Therefore, all pending claims are in condition for allowance, and such action is earnestly solicited. The Examiner is respectfully requested to contact the undersigned by telephone if such contact would further the examination of the present application.

Please charge any shortages and credit any overcharges to our Deposit Account number 02-2666.

Respectfully submitted,
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN, LLP

Date: June 18, 2007

/Vincent H. Anderson
Vincent H. Anderson
Reg. No. 54,962
Attorney for Applicants

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1026
(503) 439-8778